

Develop A Python Script

TEAM ID	PNT2022TMID50055
PROJECT NAME	Smart Waste Management using IOT Metropolitan cities

```
import time
import sys
import ibmiotf.device
import ibmiotf.application
import random
```

```
organizationID='1hdx6w'
deviceType='GAYATHRI'
deviceId='171122'
authMethod='token'
authToken='12345678'
```

```
def myCommandCallback(cmd):
    print("Command received: %s" %cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print('led is on')
    elif status=="lightoff":
        print('led is off')

    else :
        print('please send proper command')
try:
    deviceOption={"org":organizationID,"type":deviceType,"id":deviceId,"auth-
method":authMethod,"auth-token":authToken}
    deviceCli = ibmiotf.device.Client(deviceOption)
except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
    sys.exit()

deviceCli.connect()

while True:
    temp=random.randint(90,100)
    Humid=random.randint(10,100)

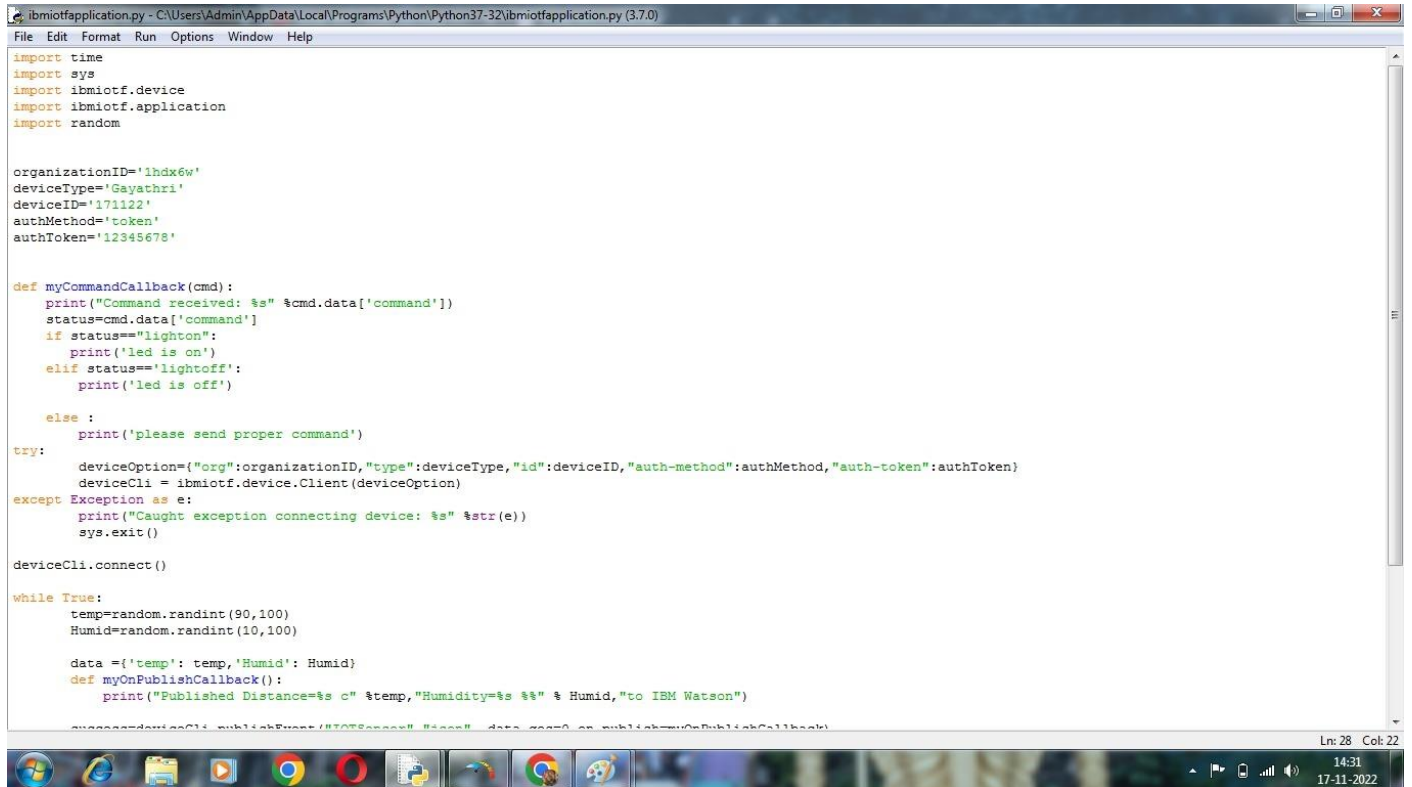
    data ={'temp': temp,'Humid': Humid}
    def myOnPublishCallback():
        print("Published Distance=%s c" %temp,"Humidity=%s %" % Humid,"to IBM Watson")

    success=deviceCli.publishEvent("IOTSensor","json",
data,qos=0,on_publish=myOnPublishCallback)
    if not success:
```

```
print("Not connected to IOTF");  
time.sleep(10)
```

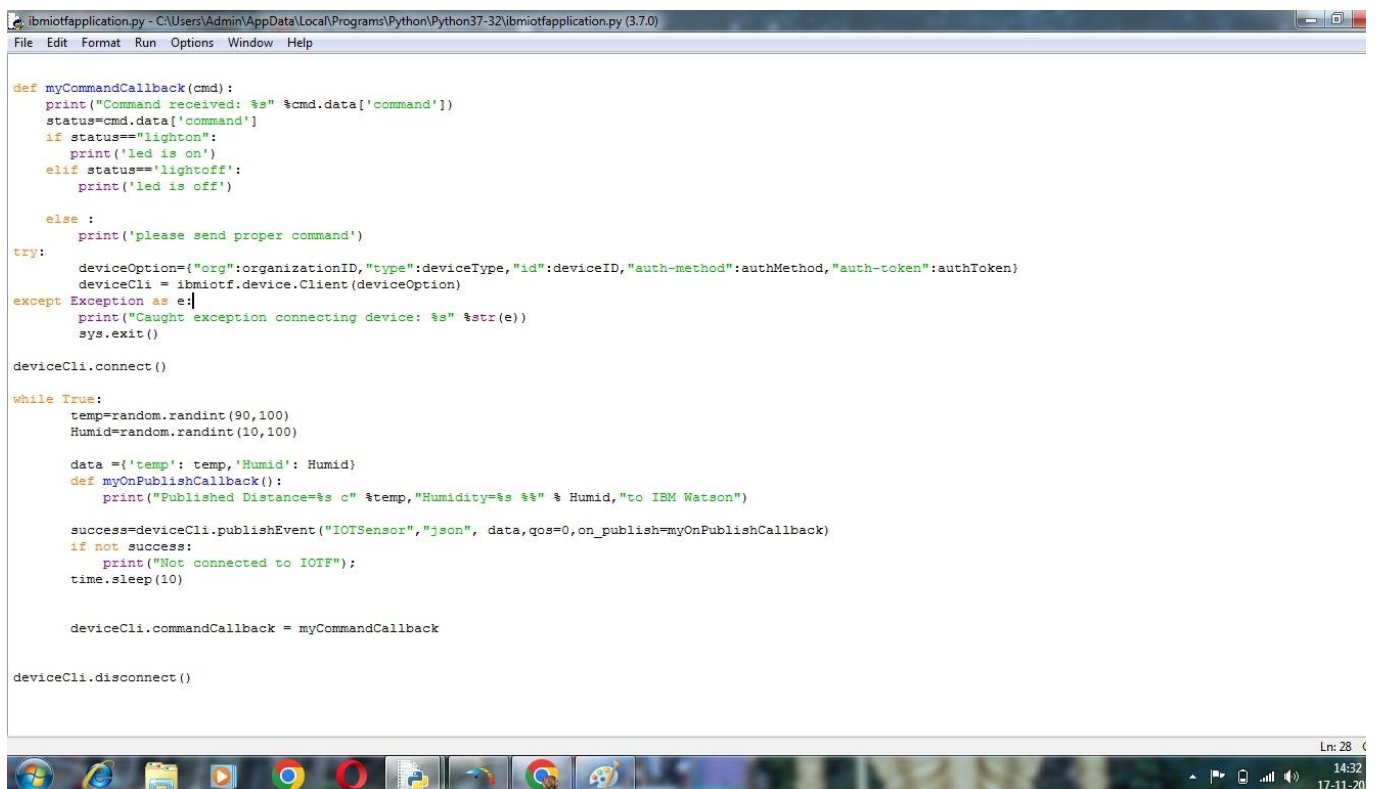
```
deviceCli.commandCallback = myCommandCallback
```

```
deviceCli.disconnect()
```



The screenshot shows a Python IDE window titled 'ibmiotfapplication.py - C:\Users\Admin\AppData\Local\Programs\Python\Python37-32\ibmiotfapplication.py (3.7.0)'. The code includes imports for time, sys, ibmiotf.device, ibmiotf.application, and random. It defines constants for organizationID, deviceType, deviceID, authMethod, and authToken. A function myCommandCallback(cmd) is defined to handle commands like 'lighton' and 'lightoff'. The main logic is in a try block where deviceOption is created, deviceCli is initialized, and deviceCli.connect() is called. A while True loop generates random temperature and humidity data and publishes it to IBM Watson. The code ends with a call to deviceCli.publishEvent.

```
ibmiotfapplication.py - C:\Users\Admin\AppData\Local\Programs\Python\Python37-32\ibmiotfapplication.py (3.7.0)  
File Edit Format Run Options Window Help  
import time  
import sys  
import ibmiotf.device  
import ibmiotf.application  
import random  
  
organizationID='1hdx6w'  
deviceType='Gayathri'  
deviceID='171122'  
authMethod='token'  
authToken='12345678'  
  
def myCommandCallback(cmd):  
    print("Command received: %s" %cmd.data['command'])  
    status=cmd.data['command']  
    if status=="lighton":  
        print('led is on')  
    elif status=="lightoff":  
        print('led is off')  
    else:  
        print('please send proper command')  
try:  
    deviceOption={"org":organizationID,"type":deviceType,"id":deviceID,"auth-method":authMethod,"auth-token":authToken}  
    deviceCli = ibmiotf.device.Client(deviceOption)  
except Exception as e:  
    print("Caught exception connecting device: %s" %str(e))  
    sys.exit()  
  
deviceCli.connect()  
  
while True:  
    temp=random.randint(90,100)  
    Humid=random.randint(10,100)  
  
    data ={'temp': temp,'Humid': Humid}  
    def myOnPublishCallback():  
        print("Published Distance=%s c" %temp,"Humidity=%s %" % Humid,"to IBM Watson")  
    success=deviceCli.publishEvent("IOTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
```



The screenshot shows the continuation of the Python code in the same IDE. It includes the myCommandCallback function, the try block for device initialization, and the while True loop. The loop now includes a call to deviceCli.publishEvent, a check for success, a print statement for 'Not connected to IOTF' with a 10-second sleep, and the assignment of myCommandCallback to deviceCli.commandCallback. The code ends with deviceCli.disconnect().

```
ibmiotfapplication.py - C:\Users\Admin\AppData\Local\Programs\Python\Python37-32\ibmiotfapplication.py (3.7.0)  
File Edit Format Run Options Window Help  
def myCommandCallback(cmd):  
    print("Command received: %s" %cmd.data['command'])  
    status=cmd.data['command']  
    if status=="lighton":  
        print('led is on')  
    elif status=="lightoff":  
        print('led is off')  
    else:  
        print('please send proper command')  
try:  
    deviceOption={"org":organizationID,"type":deviceType,"id":deviceID,"auth-method":authMethod,"auth-token":authToken}  
    deviceCli = ibmiotf.device.Client(deviceOption)  
except Exception as e:  
    print("Caught exception connecting device: %s" %str(e))  
    sys.exit()  
  
deviceCli.connect()  
  
while True:  
    temp=random.randint(90,100)  
    Humid=random.randint(10,100)  
  
    data ={'temp': temp,'Humid': Humid}  
    def myOnPublishCallback():  
        print("Published Distance=%s c" %temp,"Humidity=%s %" % Humid,"to IBM Watson")  
    success=deviceCli.publishEvent("IOTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)  
    if not success:  
        print("Not connected to IOTF");  
        time.sleep(10)  
  
    deviceCli.commandCallback = myCommandCallback  
  
deviceCli.disconnect()
```

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
C:\Users\human\AppData\Local\Programs\Python\Python37-32\ibmiotc\application.py
2022-11-17 14:27:24,348 ibmiotf.device.Client INFO Connected successfully: d:1hdx6w:Gayathri:171122
Published Distance=98 c Humidity=89 % to IBM Watson
Published Distance=91 c Humidity=21 % to IBM Watson
Published Distance=92 c Humidity=27 % to IBM Watson
Published Distance=96 c Humidity=54 % to IBM Watson
Published Distance=96 c Humidity=30 % to IBM Watson
Published Distance=91 c Humidity=93 % to IBM Watson
Published Distance=97 c Humidity=93 % to IBM Watson
Published Distance=99 c Humidity=79 % to IBM Watson
Published Distance=90 c Humidity=32 % to IBM Watson
Published Distance=98 c Humidity=38 % to IBM Watson
Published Distance=91 c Humidity=80 % to IBM Watson
Published Distance=95 c Humidity=71 % to IBM Watson
Published Distance=96 c Humidity=78 % to IBM Watson
Published Distance=96 c Humidity=41 % to IBM Watson
Published Distance=91 c Humidity=51 % to IBM Watson
Published Distance=96 c Humidity=86 % to IBM Watson
Published Distance=91 c Humidity=33 % to IBM Watson
Published Distance=95 c Humidity=26 % to IBM Watson
Published Distance=92 c Humidity=45 % to IBM Watson
Published Distance=90 c Humidity=33 % to IBM Watson
Published Distance=91 c Humidity=45 % to IBM Watson
Published Distance=94 c Humidity=14 % to IBM Watson
Published Distance=100 c Humidity=58 % to IBM Watson
Published Distance=100 c Humidity=41 % to IBM Watson
Published Distance=100 c Humidity=37 % to IBM Watson
Published Distance=95 c Humidity=92 % to IBM Watson
Published Distance=92 c Humidity=82 % to IBM Watson
Published Distance=97 c Humidity=46 % to IBM Watson
Published Distance=94 c Humidity=62 % to IBM Watson
Published Distance=98 c Humidity=82 % to IBM Watson
Published Distance=98 c Humidity=61 % to IBM Watson
Published Distance=100 c Humidity=64 % to IBM Watson
Published Distance=92 c Humidity=46 % to IBM Watson
Published Distance=92 c Humidity=95 % to IBM Watson
Published Distance=90 c Humidity=25 % to IBM Watson
Published Distance=96 c Humidity=97 % to IBM Watson
Published Distance=93 c Humidity=32 % to IBM Watson
Published Distance=93 c Humidity=10 % to IBM Watson
Published Distance=98 c Humidity=93 % to IBM Watson
Ln: 9 Col: 0
14:33
17-11-2022
```